Music Genre Classification Using Multi-Class Support Vector Machines

1. Introduction

1.1 Background

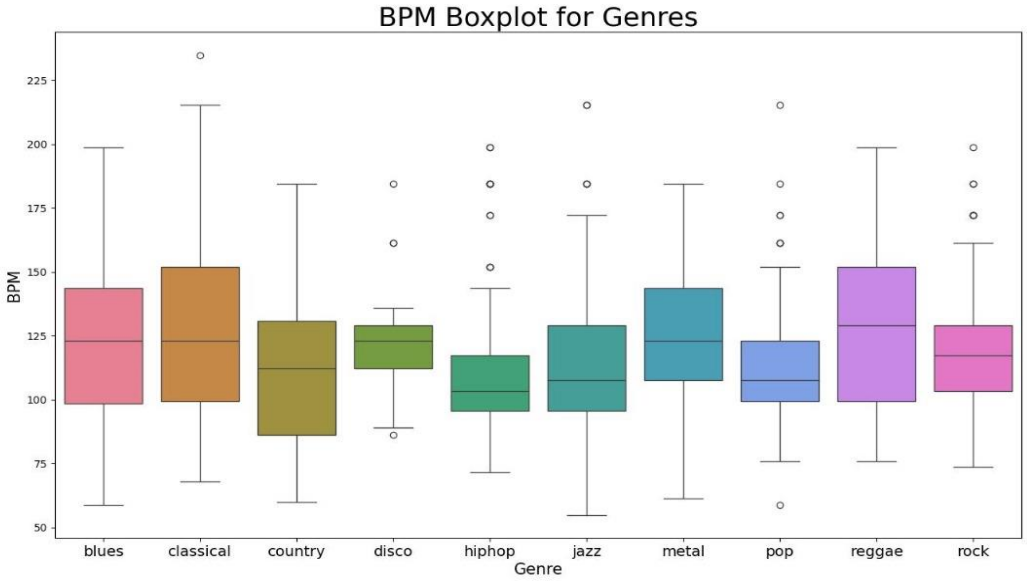
Music genre classification is a common task in audio signal processing and machine learning. It involves training a model to automatically categorize music tracks into predefined genres. Support Vector Machines (SVM) are a popular choice for such tasks due to their ability to handle high-dimensional data and nonlinear relationships.

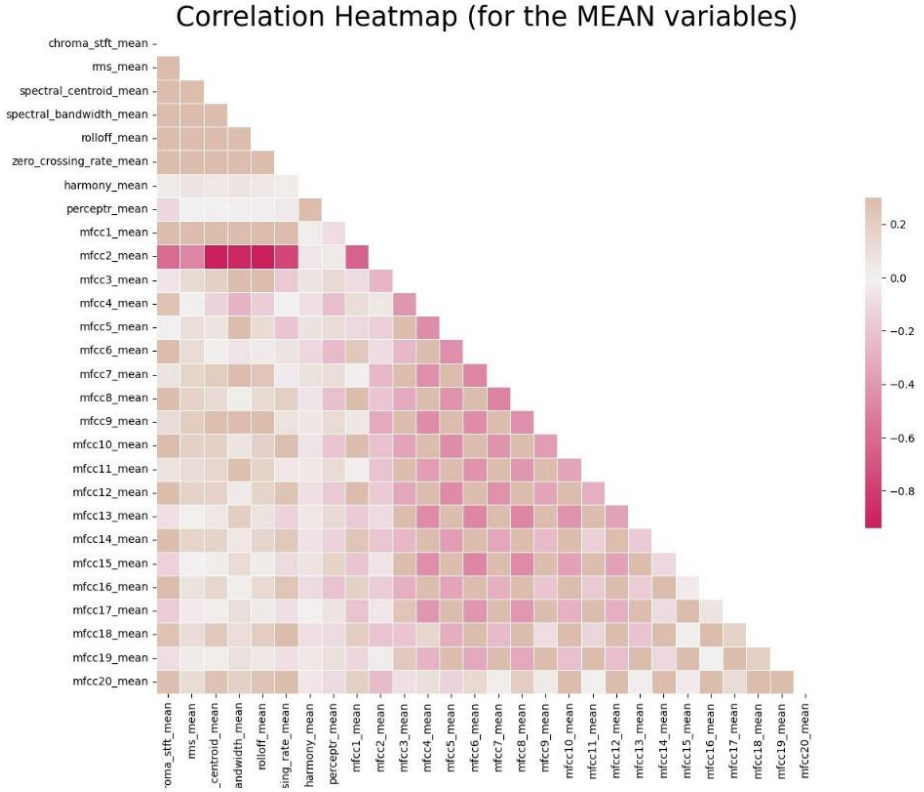
2. Data Collection and Preprocessing

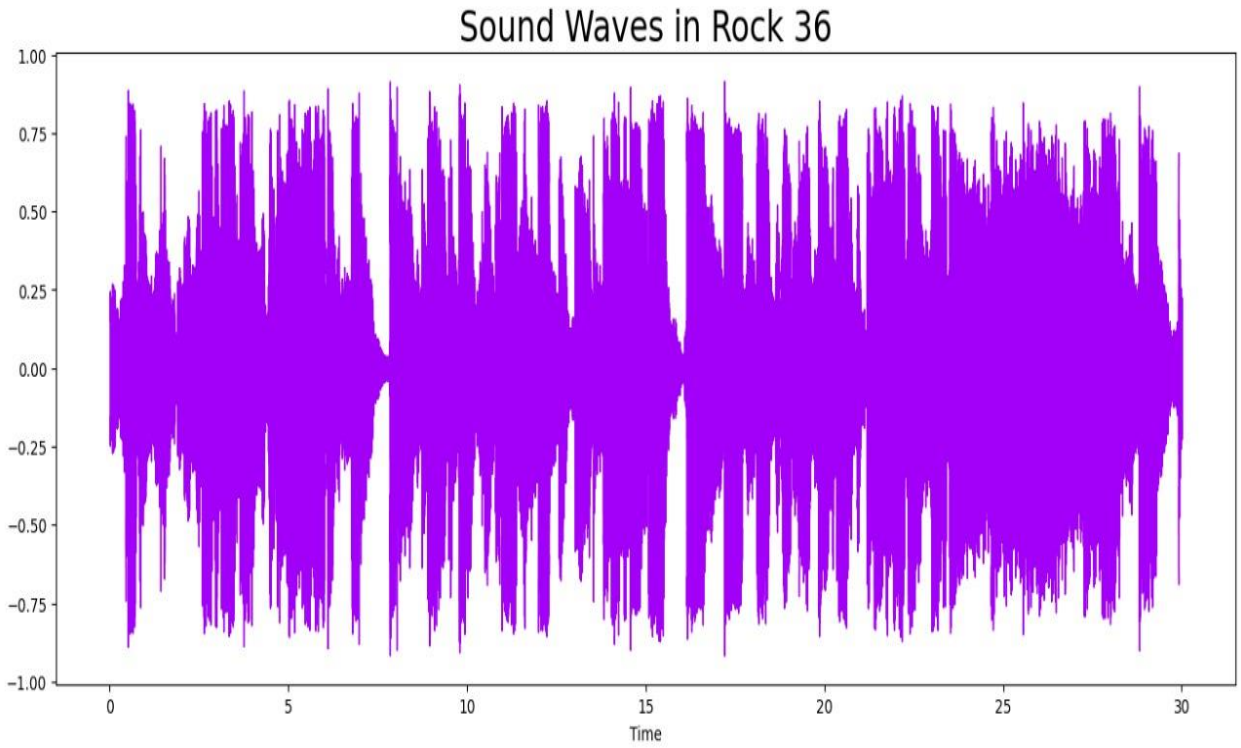
2.1 Dataset

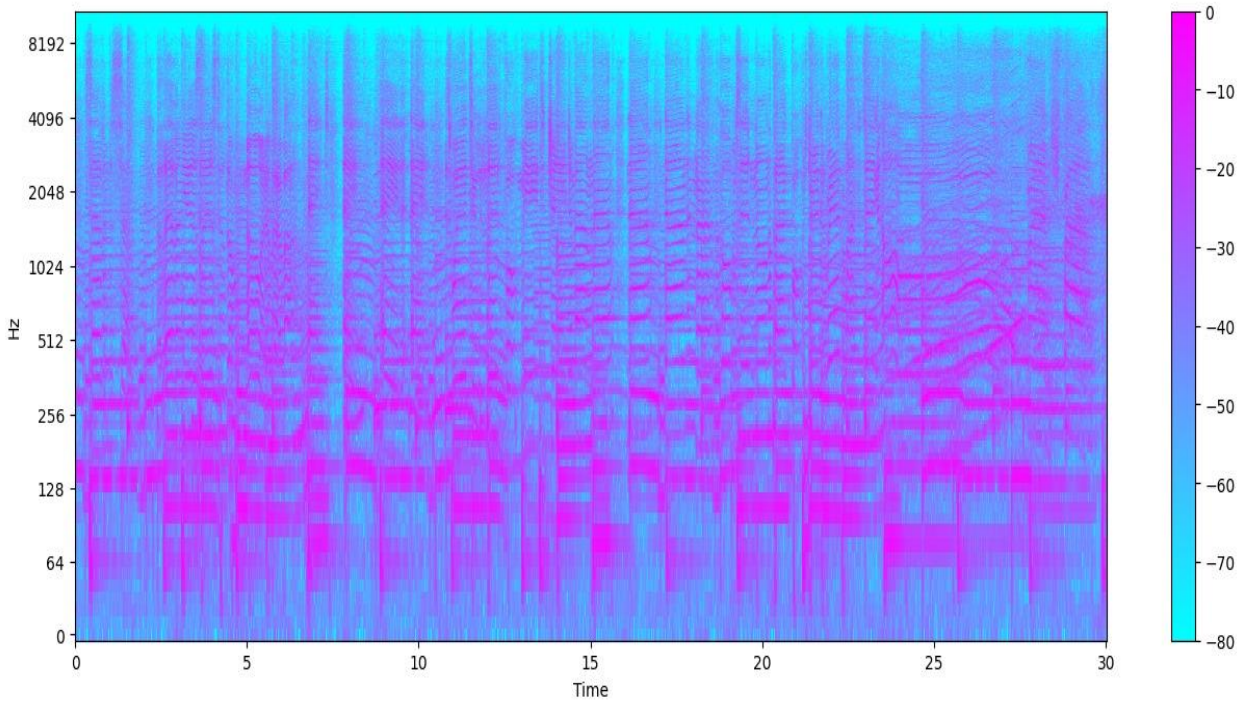
The GTZAN dataset is widely used in the field of music genre classification. It consists of 1,000 audio tracks, each 30 seconds long, spanning ten different genres.

2.2 Data Visualization


BPM Boxplot for Genres
BPM
225
200
175
150
125
100
75
50
blues
classical
country
disco
hiphop
jazz
metal
pop
reggae
rock
Genre


Correlation Heatmap (for the MEAN variables)
chroma_stft_mean
rms_mean
spectral_centroid_mean
spectral_bandwidth_mean
rolloff_mean
zero_crossing_rate_mean
harmony_mean
perceptr_mean
mfcc1_mean
mfcc2_mean
mfcc3_mean
mfcc4_mean
mfcc5_mean
mfcc6_mean
mfcc7_mean
mfcc8_mean
mfcc9_mean
mfcc10_mean
mfcc11_mean
mfcc12_mean
mfcc13_mean
mfcc14_mean
mfcc15_mean
mfcc16_mean
mfcc17_mean
mfcc18_mean
mfcc19_mean
mfcc20_mean
0.2
0.0
−0.2
−0.4
−0.6
−0.8





2.3 Data Preprocessing

Librosa, a Python package for music and audio analysis, is used for audio file processing. Features such as Chroma STFT (Short-Time Fourier Transform), RMS (Root Mean Square), Spectral Centroid, Spectral Bandwidth, Rolloff, Zero Crossing Rate, Harmony, Perceptr, Tempo, and MFCC (Mel-frequency cepstral coefficients) are extracted. These features capture essential characteristics of the audio signal.

Short-Time Fourier Transform (STFT)

The Short-Time Fourier Transform (STFT) is a technique used in signal processing and analysis to examine how the frequency content of a signal changes over time. It is particularly useful for analyzing signals that vary in frequency over time, such as speech signals or music.

Here's a brief explanation of the key concepts involved in STFT:

1. *Time-Frequency Representation:*

   - Unlike the regular Fourier Transform, which provides a frequency-domain representation of an entire signal, the STFT operates on short, overlapping segments of the signal.

   - By using short time windows, the STFT captures how the frequency content of the signal changes over time.

2. *Windowing:*

   - The input signal is divided into short, overlapping windows. Each window is typically multiplied by a window function (e.g., Hamming window) to reduce spectral leakage.

3. *Fourier Transform for Each Window:*

- The Fourier Transform is applied to each windowed segment of the signal. This results in a frequency-domain representation for each window.

4. *Time Evolution:*

- The STFT is computed for successive windows as the analysis window moves through the signal. This creates a time-evolving representation of the signal's frequency content.

5. *Spectrogram:*

- The output of the STFT is often represented as a spectrogram, which is a 2D matrix where one axis represents time, another represents frequency, and the color intensity represents the magnitude of the frequency components.

Mathematically, if $x(t)$ is the input signal, and $w(\tau)$ is the window function, the STFT $X(\omega, \tau)$ is given by:

$$X(\omega, \tau) = \int_{-\infty}^{\infty} x(t) \cdot w(t - \tau) \cdot e^{-j\omega t} \, dt$$

where $X(\omega, \tau)$ is a complex-valued function representing the frequency content at frequency ω and time τ.

The STFT is widely used in various applications, including audio signal processing, speech analysis, and image processing. It provides a time-frequency representation that is essential for understanding how the frequency components of a signal change over time.processing. It provides a time-frequency representation that is essential for understanding how the frequency components of a signal change over time.

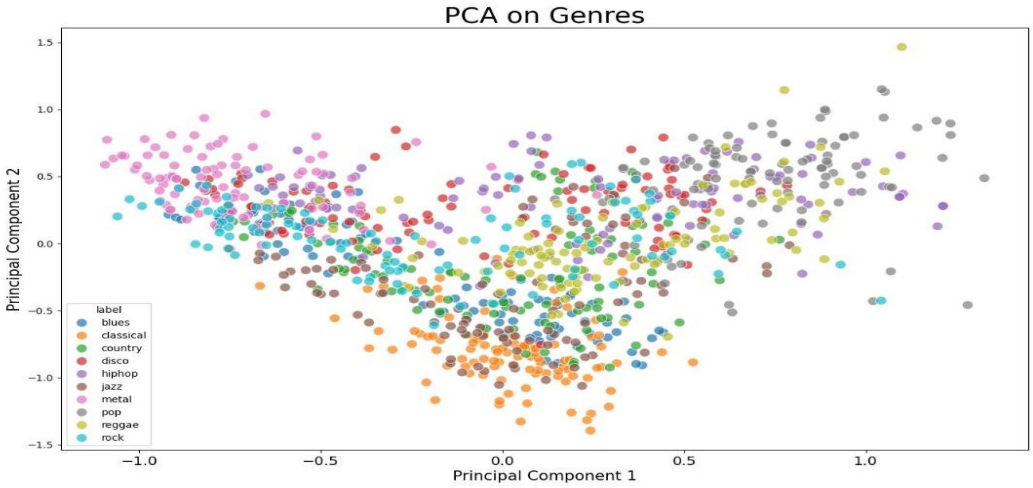
3. Model Design and Architecture

3.1 Feature Normalization

To ensure that all features have a similar scale, Min-Max scaling is applied, bringing them within a common range.

3.2 Principal Component Analysis (PCA)

PCA is used for dimensionality reduction, transforming the high-dimensional feature space into a two-dimensional space for visualization. This step aids in understanding the distribution of data points and the importance of different features.



3.3 Support Vector Machine (SVM) Classifier

SVM is employed for genre classification. It is a supervised learning algorithm that separates data points of different classes using a hyperplane in a high-dimensional space. The model is fine-tuned using a Grid Search to find the best combination of hyperparameters for optimal performance.

4. Training Process

4.1 Dataset Splitting

The dataset is divided into training, validation, and test sets. The training set is used to train the model, the validation set helps in hyperparameter tuning, and the test set evaluates the model's generalization to unseen data.

4.2 Hyperparameter Tuning

Grid Search is performed on a predefined set of hyperparameters to find the combination that yields the best model performance on the validation set.

4.3 Model Training

The SVM model is trained on the training set using the optimized hyperparameters.

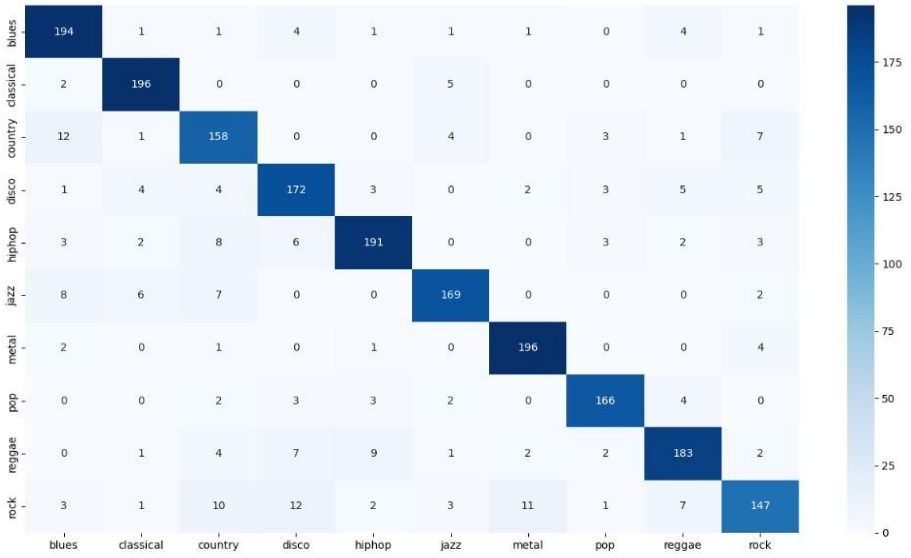
5. Evaluation Results

5.1 Model Accuracy

The trained model is evaluated on the test set, and accuracy, the proportion of correctly classified instances, is reported.

5.2 Confusion Matrix Analysis

A confusion matrix is generated to provide a more detailed evaluation. It displays the number of true positive, true negative, false positive, and false negative predictions for each genre. This allows for a deeper understanding of how well the model performs for each class.



6. Conclusion

The report concludes by summarizing key findings. It underscores the success of the SVM model in music genre classification while acknowledging the potential for further improvements.

7. Code Overview

The provided Python code implements the entire pipeline described in the report. It starts by loading necessary libraries and the GTZAN dataset. The code proceeds with audio file processing, feature extraction, data visualization, feature normalization, PCA, SVM model training, and evaluation.

Key points to note in the code:

The use of librosa for audio processing and feature extraction.

The Min-Max scaling for feature normalization.

Principal Component Analysis (PCA) for dimensionality reduction.

Implementation of a Support Vector Machine (SVM) classifier and hyperparameter tuning using Grid Search.

Evaluation metrics such as accuracy and confusion matrix for assessing model performance.

The last section of the code demonstrates real-time audio classification using the trained SVM model.